Simulating Dynamic Bandwidth Allocation on Satellite Links [*]

Alberto Gotta
DIST-Univ. of Genoa
via Opera Pia 13
16145 Genoa (Italy)
alberto.gotta@isti.cnr.it

Francesco Potortì
ISTI-CNR
via G. Moruzzi 1
56124 Pisa (Italy)
Potorti@isti.cnr.it

Raffaello Secchi
ISTI-CNR
via G. Moruzzi 1
56124 Pisa (Italy)
raffaello.secchi@isti.cnr.it

ABSTRACT

In the last years, DVB-RCS has emerged as a flexible technology offering broadband Internet access to a large community of users at a relatively low cost. At the same time, the spreading of networked multimedia applications has highlighted the need to investigate mechanisms that guarantee a certain level of Quality of Service (QoS) to the end users. In particular, the DVB-RCS standard specifies different capacity request categories to support QoS at the link layer.

We describe *Tdma-bod*, an ns-2 improvement that implements generic bandwidth-on-demand allocation in TDMA satellite systems; the patch is available as free software. This simulator has been validated through experimental tests performed on the Skyplex satellite system. Specifically, we run CBR UDP flows to measure the characteristics of the satellite link in terms of throughput and delay and to verify that the simulative model output matches the experimental dynamic throughput and one-way delay behaviour. The simulations and experiments show that bandwidth-on-demand allocation mechanisms may cause large delays when sudden variations in the incoming traffic rate occur, a behaviour typical of multimedia flows.

Categories and Subject Descriptors

I.6.5 [**Simulation and Modeling**]: Modeling Development

General Terms

Measurement

Keywords

Satellite Link Simulation, Bandwidth on Demand

[*]This work was supported by the CNR/MIUR program "Legge 449/97 (project IS-Manet) and by the European Commission under the European Satellite Communications NoE (SatNEx, IST-507052) within the 6th Research Framework Programme.

WNS2'06 , October 10, 2006, Pisa, Italy
© 2006 ACM 1-59593-508-8/06/10...$5.00

1. INTRODUCTION

In the last decade, the Digital Video Broadcasting (DVB) technology and its applications to satellite communications (DVB-S) has emerged as the leading standard to provide video streaming services through the satellite media. In 2000, to integrate IP-based services in satellite environment, DVB-S was augmented with Return Channel via Satellite (DVB-RCS), which allows the user to interact with the ISP through a bidirectional channel.

In DVB-RCS the data flows are transmitted to the satellite through a Time Division Multiple Access (TDMA) uplink from relatively small user terminals. Users' signals are demodulated, regenerated and multiplexed by the units on board the satellite and forwarded to the downlink in DVB-S format. Since all the ground stations (TT, for traffic terminals) are able to receive and transmit data flows, a DVB-RCS network allows all the traffic terminals to exchange data with each other. The inherent broadcasting capabilities of DVB-S channels are the ideal platform for multicast applications (e.g. video conferencing) and multimedia-on-demand services. Moreover, DVB-RCS provides Internet connectivity to many categories of mobile users (e.g. land, maritime) and to distant and rural locations.

In this paper we present a simulator of a DVB-RCS network based on ns-2 [7], which we call *Tdma-bod*. Since the DVB-RCS standard allows several schemes of slot assignment in a TDMA frame, Tdma-bod is designed to offer a flexible framework to the implementation of different techniques of bandwidth on demand (BoD) allocation. Among these, rate-based dynamic capacity (RBDC) has been validated through the comparison with actual traffic measurements.

Our simulative analysis confirms previous findings [3] regarding experimentally observed interactions between the network transport layer and the satellite medium access control (MAC) layer. Specifically, we observed that the transmission delay dramatically increases when the rate of traffic incoming to a TT varies rapidly. This behaviour is due to the *allocation delay*, that is, the time interval between the TTs requesting a given bandwidth and the system assigning it. Indeed, if the input rate increases in a time shorter than the allocation delay, the excess traffic is buffered at the traffic terminal, thus increasing the queuing delay.

Tdma-bod is an extension of the ns-2 satellite platform [6, 9] supporting BoD allocation algorithms for DVB-RCS. Tdma-

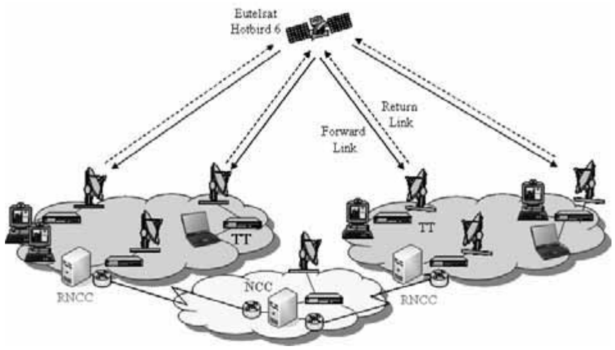


Figure 1: DVB-RCS Network Architecture

bod makes it possible to analyse the performance of network protocols in satellite environment using dynamic bandwidth demand-assignment methods. The architecture of the simulator and some of its allocation strategies are inherited from Fracas [10]. Thanks to ns-2, Tdma-bod overcomes the main limitation of Fracas, which is not able to simulate closed-loop protocols, such as the TCP. Tdma-bod is available as free software at http://wnet.isti.cnr.it/software/tdma-bod: we believe that software developed as part of research activity should be released with a *free software license*, because research results should be made available for use by anyone, for any purpose and be freely modifiable, in order to further knowledge and usefulness [12].

The rest of the paper is organised as follows. Section 2 discusses the features of a DVB-RCS satellite platform. Section 3 describes the current status of ns-2 satellite support, the architecture of the Tdma-bod simulator and the algorithms for dynamic bandwidth allocation that are implemented. Section 4 describes to the ns-2 user interface. In Section 5 we compare the results provided by simulator with real measurement on a DVB-RCS network, in terms of throughput and one-way delay.

2. DVB-RCS SYSTEM

Digital Video Broadcasting with Return Channel via Satellite (DVB-RCS) [5] has emerged as the platform of choice to integrate the satellite broadcasting capabilities with the Internet infrastructure. The IP/DVB architecture defines the basic set of functionalities required for user interactivity across a satellite DVB-S distribution system. Basically, IP packets are first encapsulated into an MPE (MultiProtocol Encapsulation) structure containing the source/destination MAC addresses of the traffic stations involved in the data transfer. Then, MPE packets are delivered using a transport stream made up of 188-bytes MPEG-2 cells.

Fig. 1 shows the main elements of our DVB-RCS network. The network is divided into regions, corresponding to the geographic areas covered by each satellite spot beam. The group of traffic terminals (TTs) that belong to a specific region is monitored and controlled by a centralised authority, called Regional Network Control Center (RNCC), which is connected through a leased line with the NCC, a higher level network management system. The Multi-Frequency Time Division Multiple Access (MF-TDMA) scheme [5] is used to share the satellite uplink capacity. In this way, the time-frequency domain is organised into a hierarchical structure, which consists of superframes (the largest containers), multiframes and frames.

In order to establish the time slot assignments for the traffic terminals, the RNCC broadcasts periodically a burst time plan (BTP) message to its group of TTs. The DVB-RCS standard describes several methods of bandwidth allocation, which designers can combine to reach the desired QoS objectives:

Constant Rate Assignment (CRA). The link resources are negotiated at the beginning of the transmission and are maintained for all the duration of the connection. Thus, this scheme does not require bandwidth-consuming dynamic signalling from TTs to RNCC. Resources are allocated even when not used, so this method may lead to waste of usually precious satellite bandwidth.

Rate Based Dynamic Capacity (RBDC). In this DAMA scheme, a TT periodically submits to the RNCC a capacity request message based on measurement of the local incoming traffic rate. Every explicit request overrides the previous one and new requests are submitted only if needed. RBDC has two parameters: *ceiling rate* ($\mathrm{RBDC}_{\mathrm{max}}$) and *instantaneous rate*. The ceiling rate is established during the call setup using a method similar to CRA and, taking this threshold into account, the scheme tries to guarantee the requested rate. A request remains effective as long as it has not timed out after a certain number of frames.

Volume Based Dynamic Capacity (VBDC). In this scheme, the TT dynamically signals the data volume required to empty its buffer. The scheduler assigns the capacity according to the requests that terminals send when more traffic is queued at their input. Since the BoD controller cannot guarantee the requested bandwidth to the TTs, this access scheme is more suitable for best effort traffic flows.

A VBDC variant, known as *guaranteed VBDC*, has been proposed that guarantees a minimum amount of data per frame to each TT.

Free Capacity Assignment (FCA). In this scheme, the capacity is assigned by the BoD controller to the TTs without any explicit request by the TTs in order to satisfy utilisation and fairness criteria. This scheme could in principle improve the MAC scheduling performance especially under low load conditions and reduce the jitter in traffic flows.

3. SIMULATOR ARCHITECTURE

In order to realize a simulative framework to study the performance of IP-based applications over dynamic bandwidth satellite links, we chose the ns-2 [7] discrete event simulator as a starting point. Ns-2 is extremely diffused in the networking community for its characteristics: it is based on an open architecture, is suitable to very different purposes, from complex protocol analysis to education, and it is generally easy to extend and modify with new features. ns-2 is the result of a project featuring the participation of both industry and academia, which is freely available for research purposes.

3.1 Satellite Networking in ns-2

This section describes the status of satellite networking in ns-2[8]. In particular, we summarize the extensions that allow to ns-2 to model geostationary "bent-pipe" satellites, regenerative geostationary satellites with on-board processing payloads and polar orbiting LEO constellations.

These satellite models are principally aimed at studying networking aspects of satellite systems such MAC, link-layer, routing and transport protocols. Though the detailed simulation of a satellite network would require to account for many aspects, such as radio frequency characteristics, error detecting and correcting codes, second-order orbital effects, from a networking perspective many of these features have a minimal impact on protocol performance and can be abstracted out by adopting appropriate channel models. Hence, the approach adopted in ns-2 is to create a framework to study transport, routing, and MAC protocols in a satellite environment consisting of geostationary satellites or constellations of polar-orbiting low-earth-orbit (LEO) satellites. As usual, users may extend these models to provide more detail at a given layer.

3.1.1 Geostationary satellites

Geostationary satellites orbit the Earth at an altitude of 36000 km above the equator. The position of the satellites is specified in terms of the longitude of the nadir point (satellite point on the Earth's surface)

Two types of geostationary satellites can be modelled. The traditional *"bent-pipe" satellites*, that merely repeat on the downlink channel the signal received from the uplink channel, and *regenerative satellites* with baseband processing capability, which are able to regenerate the digital signal and to perform fast packet switching on-board the spacecraft. In the simulations, the regenerative satellites can be modelled as ns-2 nodes supporting packet classifiers and routing agents.

The key enhancement of satellite extensions with respect to common ns-2 delay links is the capability to simulate MAC protocols. Users can define many terminals at different locations on the Earth's surface and connect them to the same satellite uplink and downlink channels, and the propagation delays in the system (which are slightly different for each user) are accurately modelled. In addition, the uplink and downlink channels can be defined differently (perhaps with different bandwidths or error models).

3.1.2 Low Earth Orbiting satellites

Polar orbiting satellite systems, such as Iridium [13], can be simulated with ns-2. The simulator models constellation of satellites that follow purely circular trajectories. In a constellation, the satellites that belongs to the same orbital plane are spaced equally along the orbit, and multiple orbital planes with relative fixed phase between them are arranged.

In order to define the satellite constellation that the user is requested to specify the satellite altitude, the number of orbital planes, the number of satellites per plane and the inclination of plane with respect to the Earth axis. By means of the inclination parameter, the user can also set up retrograde orbits.

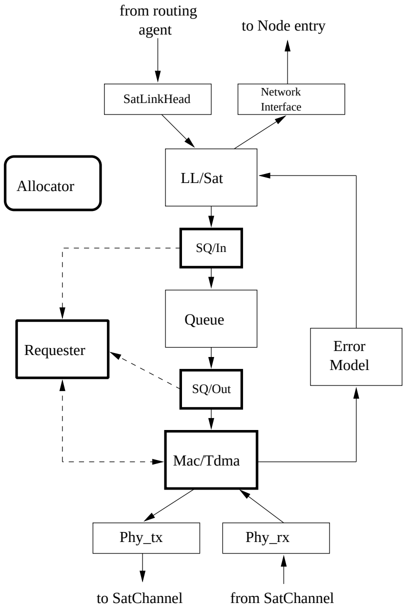


Figure 2: Architecture of Simulative Framework and Main Components. Bold frames show added or modified components

For polar orbiting constellations interplane, intraplane and crossseam intersatellite links (ISL) can be established. The *intraplane* ISLs is a connection between satellites which belong to the same orbit plane and are never deactivated or handed off. The *interplane* ISLs are established between satellites of neighboring co-rotating planes. These links are deactivated near the Earth poles because the antenna pointing mechanism cannot track these links in the polar regions. Finally, the *crossseam* ISLs are intersatellite links subjected to hand-off, which for instance happens when orbit planes are counter-rotating.

A ground to satellite link (GSL) is a connection established between the terminal and satellite. While geostationary satellite links are static, GSLs for LEO constellations are periodically handed off. Periodically, the satellite terminal checks the elevation angle of satellite. If the elevation angle drops below a threshold (the elevation mask), the terminal searches for a new satellite above the elevation mask.

3.2 The Tdma-bod extension to ns-2

Our main contribution [1] concerns the modification of the MAC layer block to support dynamic bandwidth allocation mechanisms and IP packets fragmentation and frame encapsulation. A substantial programming effort has been devoted to the adaption of Tdma-bod to the standard ns-2 satellite platform. In the following we focus only on the case of geosynchronous bent pipe satellite. This implies that the propagation delay can be considered constant and large as compared to delay in terrestrial networks.

Fig. 2 depicts the block diagram of the Tdma-bod architecture, which has been derived from ns-2's satellite link. The diagram shows the functional blocks of a satellite terminal from networking layer down to the physical layer, including layer-II specific functions (LL/Sat), output queue (Queue), MAC layer functions (MAC/Tdma) and physical layer (Phy).

Two separate objects, respectively the *Requester* and *Allocator*, have been introduced to handle the negotiation of satellite bandwidth. Since most of the additions involve packet treatment and processing, we implemented the main parts of our simulator in C++. The Otcl interface, described in section 4, gives access to the main configuration parameters and allocation functions.

3.3 Bandwidth on Demand block

The simulator consists of three main components depicted in the diagram: the MAC/Tdma agent, the Requester agent and the Allocator agent. The Requester agent, integrated into each satellite terminal, is responsible for calculating the bandwidth requests on the basis of the traffic monitored at the queue and to submit the request to the master controller. In order to monitor the queue, two traffic snoop agents (*SnoopQueue/In* and *SnoopQueue/Out*) record all the incoming and outgoing traffic, as soon as the packet are enqueue or dequeued. The Allocator agent is in charge of collecting the requests of all the active terminals, assembling the BTP (implemented as a vector containing the frame slot assignments together with the referenced terminal) and broadcasting it to all the terminals.

The core of the simulator is a procedure inside the MAC/Tdma block that is called periodically at the beginning of each time slot. In this procedure, the MAC/Tdma agent first decides whether the slot is assigned to its traffic terminal by reading the last received BTP. When the MAC/Tdma agent decides it is time to transmit, it extracts from the queue enough data to fill one time slot and forwards such data to the physical layer. While encapsulating or decapsulating the packets to Mpeg2 cells, the Mac/Tdma agent must also perform packet fragmentation and reassembly operations before forwarding information. Furthermore, this agent is in charge of invoking, every fixed number of TDMA frames (*resource allocation period*), the Requester *dorequest()* function to issue a bandwidth request. The actual implementation of this function depends on the allocation scheme in use. However, the *dorequest()* function always returns an integer parameter, which is conveyed to the Allocator agent, overriding the previously submitted request.

The Allocator agent collects the requests of all the active terminals and stores them in a vector that is passed to the function *allocation()*. The implementation of *allocation()* is strictly related to the employed scheduling scheme. The user may provide his customised redefinition of this function using OTcl interface. Through this function, the allocator examines the vector of requests and produces the BTP, which describes the scheduling of slots applied in a subsequent frame.

The time between the submission of request and its fulfilment through the BTP is called the *allocation delay*. The allocation delay is given by the propagation delay from Requester agent to Allocator agent, the processing delay, the propagation delay of BTP and an additional period (safe frame period) that ensures that all the terminals received the BTP on time. During the allocation delay the Requester agents can submit other requests if new packets are buffered. The allocation delay is controlled through an Otcl variable; it strongly affects the performance and the stability of the satellite network.

Even though Tdma-bod is conceived to support many allocation categories, including the four DVB-RCS allocation scheme discussed in section 2, in the following we describe in detail only the allocation schemes that we used to validate the simulator, which are a satisfactory approximation of the behaviour of the actual Skyplex satellite testbed. The RBDC model is particularly suited for delay-sensitive applications, such as interactive traffic, which require a reduced access delay and a low level of jitter.

3.4 RBDC Requester

A typical model of rate-based request assumes the following form [2]:

$$R_{\mathrm{req}}(t) \quad = \quad R(t) \; + \; \frac{q(t)}{T_s} \tag{1}$$

where $R(t)$ is the mean traffic rate incoming to TT over an interval $[t - T_r, t]$, $q(t)$ is the queue length at time t and $1/T_s$ is a proportionality factor. The rationale of (1) is that when a bandwidth equal to the right part of (1) is allocated, the scheduler provides enough bandwidth to serve the flow and empty the queue in T_s seconds.

Looking at the TT queue as an input-output system, the expression (1) can be interpreted as a proportional-integral controller of output rate, with $R(t)$ and $q(t)$ representing the proportional and the integral parts respectively. Although the integral control guarantees the queue length stability, it may cause long settling delay especially for slowly variable input rates. However, in our implementation stability does not rely uniquely on the term $q(t)/T_s$, because the scheduler provides a share of bandwidth R_{min} in addition to the requested bandwidth. This behaviour, in accordance with the CRA paradigm, aims at quickly emptying the queue when traffic rate variations are smooth. Note that this method guarantees a minimum share of bandwidth R_{min} (even if unused) to each active terminal, which considerably reduces the initial access delay.

We chose the interval T_r equal to the resource allocation period. Besides facilitating the implementation of the Requester, this choice makes it unnecessary the use of low-pass filtering to absorb steep input rate variations, since the resource allocation period is typically quite large with respect to the transmission delay. Thus, indicating with x_k the amount of data arrived at the queue during the k-th resource allocation period, we have a request in slot per frame

$$R_k = \min\{\lceil \frac{x_k}{L_s n_s} \rceil, \; R_{max}\} \tag{2}$$

where L_s is the size of a slot and n_s is the number of frames between requests. Notice that the request R_k is bounded by the R_{max} value negotiated when the TT logs onto the satellite network.

Table 1: MAC/Tdma agent configuration variables

variable name	meaning
`max_slot_num_`	Slots per frame (excluded preamble)
`slot_packet_len_`	Size of a slot [bytes]
`num_frame_`	Allocation period [frames]
`bandwidth_`	Satellite link capacity

3.5 Proportional Allocator

The objective of the scheduler is to assign bandwidth resources following fairness criteria. The mechanism here described uses a combination of CRA and bandwidth-on-demand strategies. According to our scheme, the scheduler first reserves a predetermined number of slots per frame (R_{min}) to each active station, then assigns the remaining slots on the basis of requested bandwidths.

The scheduling procedure, invoked once every n_s frames, takes the vector of requests as input and produces the vector of allocations as output. The counter $d(i)$ is associated to the i-*th* active TT: it represents the number of slots due to that TT. The counter $d(i)$ is incremented, at the time of scheduling, by the slot request $R_k(i)$ of i-*th* station if the sum of requests is less that the number of residual slot; otherwise, it is incremented proportionally to the request. In the latter case the increment is given by

$$\Delta d(i) = \frac{(N_s - C_s N)}{\sum_{j=1}^{N} R_k(j)} R_k(i) \qquad (3)$$

where N_s is the number of slots per frame, N is the number of active stations and C_s is the number of slot per frame allocated in CRA to each station.

The scheduler cycles among all the active stations in round-robin fashion until all the slots in the frame are assigned or all the requests are satisfied. A request is considered satisfied if $d(i) \leq 0$. Then, if the scheduler finds that the i-*th* station has $d(i) > 0$ and some slots are still unallocated, it assigns a slot to that station, decrements the counter $d(i)$ by one and switches to the next station. The scheduler uses the vector $d(i)$ as a memory of the backlog of each traffic terminal, so when a request cannot be satisfied in the current frame because no resources are available, it will be satisfied in the next frame, when the scheduler resumes slot assignment from where it had left.

The procedure just described may finish without all the slot being assigned in a frame because, for instance, the sum of the bandwidth requests of all active TT turned out to be lower than the total frame capacity. In order not to waste bandwidth resources, the remaining slots can be managed in FCA mode. In our implementation, we provided the possibility to allocate the remaining slots proportionally to the requested bandwidth. However, for comparison with the real satellite testbed, we disabled this feature.

4. SIMULATOR INTERNALS

4.1 OTcl Interface

In this section we briefly discuss the OTcl interface. Table 1 illustrates the configuration variables of the MAC/Tdma object, which should be initialised before creating a satellite terminal. The fragment of OTcl code below shows how to configure the terminal node through the procedure *node-config()*, which is consistent with the standard ns-2 procedure:

```
set chan      Channel/Sat
set bw_up     2.112Mb
set bw_down   38.016Mb
set phy       Phy/Sat
set mac       Mac/Tdma
set ifq       Queue/DropTail
set qlim      1500
set ll        LL/Sat

$ns node-config \
  -satNodeType terminal  -llType $ll \
  -ifqType $ifq          -ifqLen $qlim \
  -macType $mac          -phyType $phy \
  -channelType $chan     -downlinkBW $bw_down
```

We specified the channel class (Channel/Sat) together with uplink and downlink bandwidth, the physical layer class (Phy/Sat) and MAC class (Mac/Tdma), the queuing discipline (Queue/DropTail) and queue size, and link layer class (LL/Sat).

After being configured, the satellite terminal can be created and connected with the geostationary satellite node. The following set of commands sets up the bidirectional connection between satellite node and terminal node:

```
set ter [$ns node]
$ter add-gsl geo $ll \
        $ifq $qlim \
        $mac $bw_up $phy \
        [$sat set downlink_] \
        [$sat set uplink_]
```

Finally, the Allocator and Requester objects are created and installed into terminals, through the *install-allocator* and *install-requester* procedure respectively. Note that a Requester object is needed for each terminal, but only one Allocator object is needed for the whole satellite network:

```
set alc [$ter install-allocator Allocator/Proportional]
set req [$ter install-requester Requester/RDBC]
```

The definition of Requester and Allocator must specify the subclass of the objects. The object name can be used at run time to dynamically change the properties of Allocator and Requester. Specifically, the `request_` OTcl variable of Requester object gives access to the request submitted by the terminal to the allocator.

4.2 Output Trace Format

Tdma-bod is provided with tracing support relying on the ns-2 callback mechanism. The trace allow to keep track of requests, bandwidth allocation and queue state. Here we give a sample of trace:

```
<0> 0.267758 reques 0 0 0 0 0
<0> 0.267758 nxtall 0 0 0 0 0
```

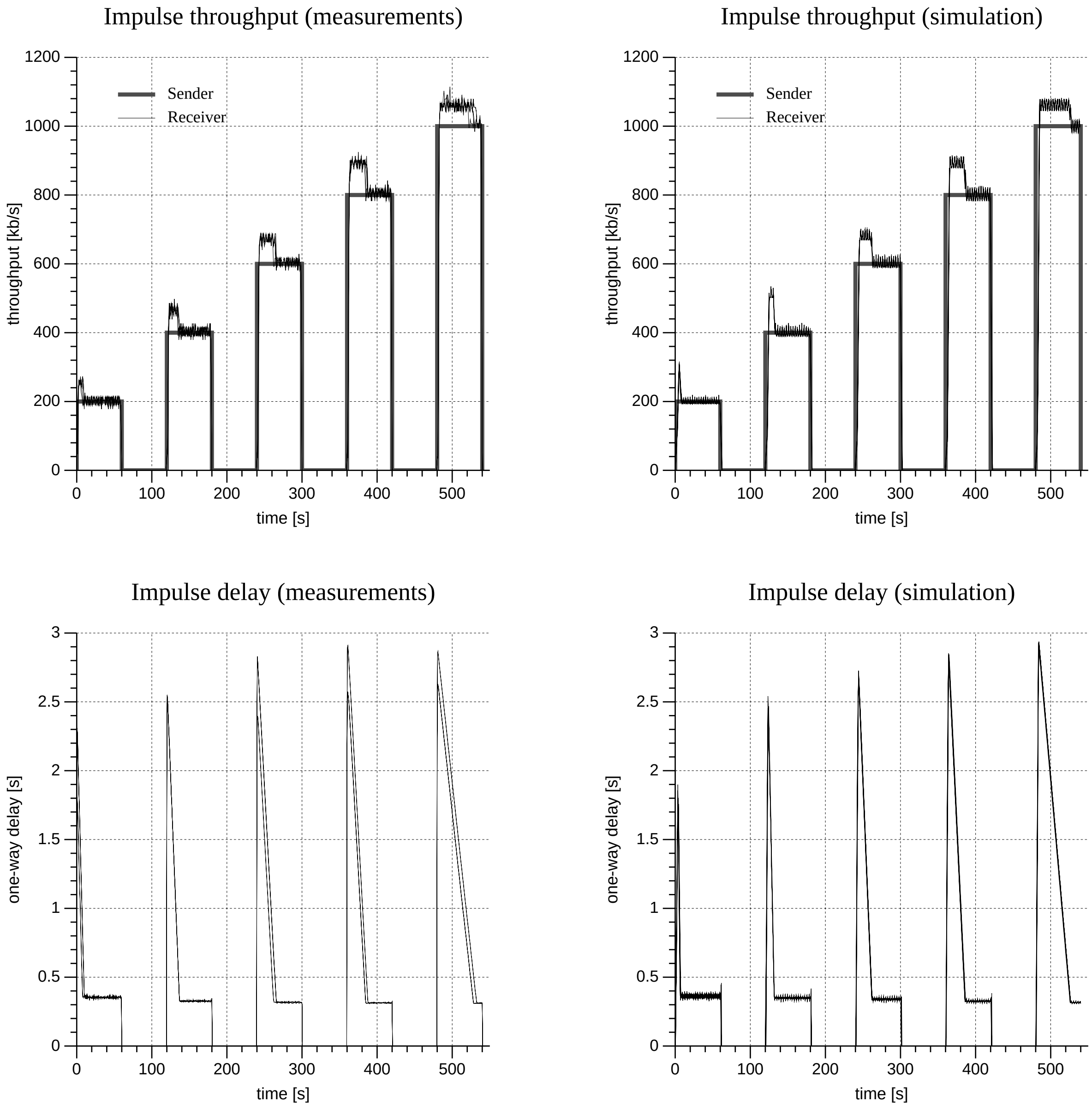
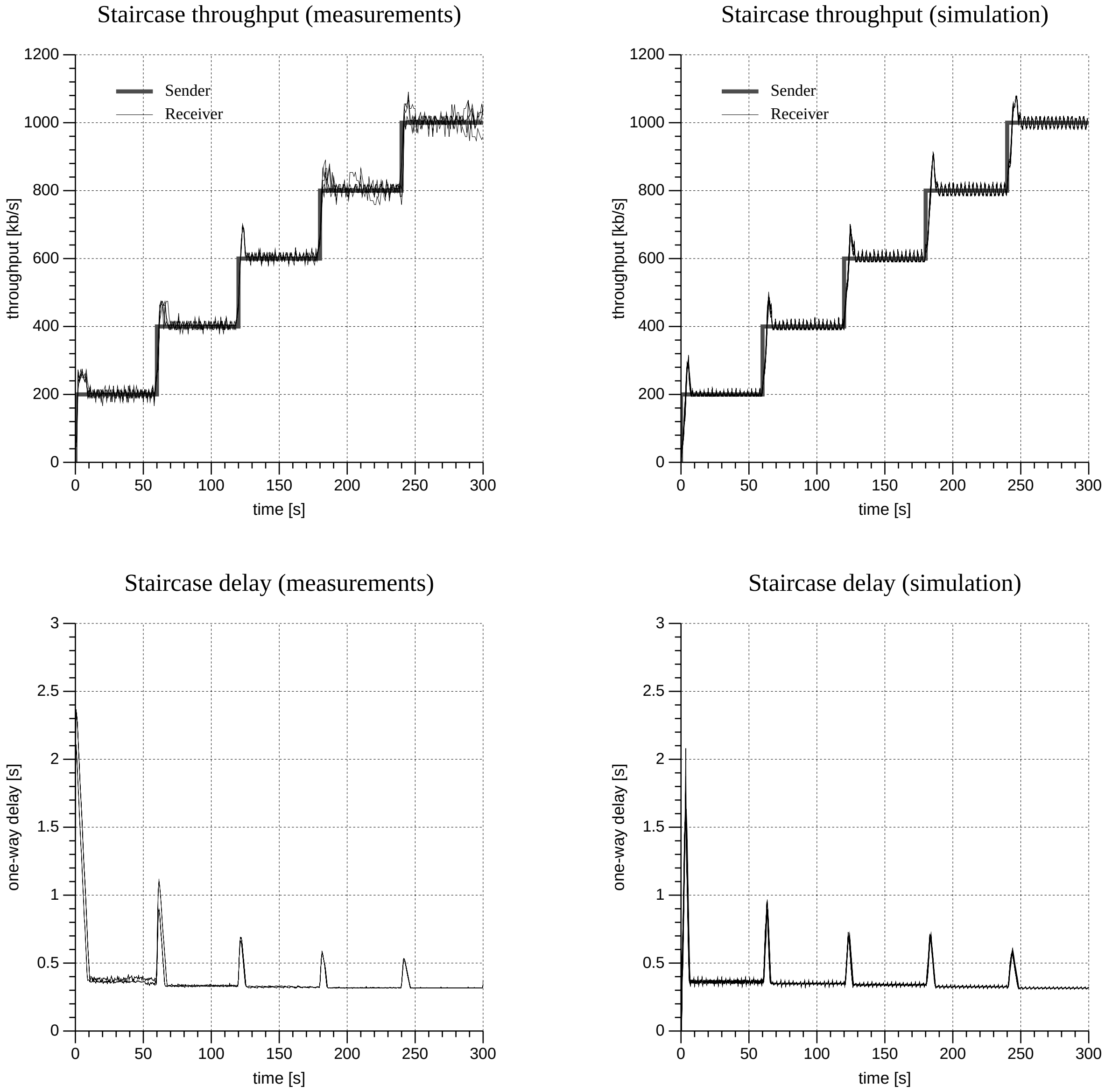


Figure 3: Throughput and one-way delay with an impulse traffic pattern


Staircase throughput (measurements)

Staircase throughput (simulation)

Staircase delay (measurements)

Staircase delay (simulation)

Figure 4: Throughput and one-way delay with a staircase traffic pattern

```
<0> 0.267758 nxtslt [N][N][N][N][N]
<0> 0.267758 curall 0 0 0 0 0
<0> 0.267758 curslt [N][N][N][N][N]
<0> 0.267758 reqpar 0 0
```

The first column reports the terminal identifier (between angulars), the second the current time and the third is a label that specifies the entry type. We supply six labels. Entries labelled `curall` and `curslt` list the number of slots assigned per station and the structure of the current frame. Entries labelled `nxtall` and `nxtslt` do the same for the frame currently handled by Allocator, which will be used after an allocation delay. Entries labelled `reques` indicates the requests of all active TT. Entries labelled `reqpar` are written by the Requester and list the mean input rate at the queue and buffer state.

5. VALIDATION

The dynamic allocation module can emulate any dynamic bandwidth allocation policy applied to a DVB-RCS satellite system or, more generally, a generic TDMA based satellite system. In order to validate it against a real system, we have focused our attention on Skyplex Data from Eutelsat, an IP-based satellite platform based on DVB-RCS technology [4]. The Skyplex Data system from Eutelsat is based on some DVB-RCS features: uplink signals from many different geographic locations are collected, packaged on-board the satellite and delivered to their destinations. The satellite link is accessed through a Skyplex satellite terminal connected to a 90 cm satellite dish antenna, directed towards a transponder of the geostationary satellite HotBird6, which operates in the Ka band. The Skyplex network has four Regional Networks sharing an RNCC traffic terminal, which is located in the area where the four spot beams overlap. This traffic terminal brings together the functionalities of NCC and RNCC.

The satellite uplink may be in SCPC (Single Channel Per Carrier) or in TDMA mode, but capacity sharing is available only in the latter case. TDMA mode is organised in frames, multiframes and superframes. Specifically, a multiframe is made up of 8 frames, each containing 6 time slots. Every time slot is composed by 8 cells (MPEG-2 packets), so each frame carries 48 cells. Since the total duration of a multiframe is 273.4 ms and the size of each cell is 188 bytes, the global channel capacity is 2.112 Mb/s with a bandwidth assignment granularity of 44 kb/s.

The BTP is delivered to the TTs by the NCC once per superframe (that is, every 820 ms), which includes 3 multiframes.

We used two laptops running Linux 2.6, located at the ISTI research laboratory of Wireless Networks (WnLab) in Pisa and at the CNIT Research Unit in Pisa (Italy), respectively. The hosts communicate in a point-to-point topology as sender and receiver at the two end points of the satellite pipe. We used the Network Time Protocol (NTP) to synchronise the clocks on the two nodes with an accuracy of 10 ms for the delay estimation. We used BRUTE [11] to generate high precision UDP traffic with different rates and variable traffic pattern.

A total of eight simulation runs have been performed, each with a different offset of the bandwidth request with respect to the DVB superframe, thus covering all possible cases.

The top graphs in Fig. 3 show the response of the Skyplex network to impulses of UDP traffic in terms of throughput and delay. Graphs are depicted for both ns-2 simulation results and real trials. The response to an offered traffic impulse is a throughput transient with amplitude R_{min} lasting for T_a, followed by a throughput transient with amplitude $R + R_{min}$ ending at T_t, after which the throughput is equal to the offered traffic. All the following results are valid for $R > R_{min}$.

The duration of the throughput transient T_t depends on the impulse amplitude. In fact, the moment the source starts sending packets at rate R, the service rate allocated to the TT is R_{min} (the minimum allocation assigned in CRA mode), meaning that the queue at the TT fills up at a rate $R - R_{min}$. After an allocation delay of T_a, the scheduler assigns the bandwidth $R + R_{min}$ and the queue empties at a rate R_{min} (that is, the difference between the output and the input rates). The queue length reaches its maximum $T_a(R - R_{min})$ after a time equal to the allocation delay T_a. Since the total duration T_t of the transient is the sum of the allocation delay and the time required to empty the queue, we have

$$T_t = T_a + T_a \frac{R - R_{min}}{R_{min}} = T_a \frac{R}{R_{min}}, \qquad (4)$$

which means that length of the transient is proportional to the offered traffic rate.

Bottom graphs in Fig. 3 show results for the one-way delay. As expected, the peak amplitudes during the transient weakly depend on transmission rates. Indeed, the maximum queuing delay T_d is given by the ratio between the maximum queue size and the offered traffic rate, which yields

$$T_d \ = \ \tau + T_a \, (1 - \frac{R_{min}}{R}), \qquad (5)$$

where τ is the satellite system latency, which is 310 ms in our measurements.

Fig. 4 depicts the throughput and one-way delay with a staircase traffic pattern. The duration of each step is 60 s and the increment ΔR is 200 kb/s per step.

Analogously to the case of impulse offered traffic, we have traffic transients of duration

$$T_t = T_a + T_a \frac{\Delta R - R_{min}}{R_{min}} = T_a \frac{\Delta R}{R_{min}}, \qquad (6)$$

and delay spikes whose height is

$$T_d \ = \ \tau + T_a \, \frac{\Delta R - R_{min}}{R}. \qquad (7)$$

6. CONCLUSIONS

This paper describes Tdma-bod, a simulator of TDMA satellite link based on ns-2 that allows investigation of a large class of BoD schemes, including those outlined in the DVB-RCS standard. We validated the simulator through measurements performed on the Skyplex network, which implements a variant of the RBDC allocation mechanism, by comparing measurements and simulation output. The statistics considered were throughput and one-way delay for two traffic profiles, namely staircase and impulses. The simulator was able to accurately reproduce the dynamics of the real system in both cases. Incidentally, the simulations highlight that BoD algorithms are responsible for the large delay (several times the propagation delay) that may occur in response to abrupt variations of input rate. Future work includes integrating additional allocation mechanisms and analysing how bandwidth on demand influences the performance of TCP and multimedia traffic.

7. REFERENCES

[1] Ns2 simulator for satellite platform. available online - http://votos.isti.cnr.it/simulator.htm.

[2] Celandroni, N., Ferro, E., and Potortí, F. Comparison between distributed and centralised demand assignment TDMA satellite access schemes. *International Journal of Satellite Communications 14*, 2 (Mar.–Apr. 1996), 95–111.

[3] D.Adami, S.Giordano, M.Pagano, and R.Secchi. Modeling the Behavior of a DVB-RCS Satellite Network: an Empirical Validation. in proceedings of Het-Net 2005.

[4] E. Feltrin, E. Weller, E. Martin and K. Zamani. Design, Implementation and Performance Analysis of an On Board Processor-Based Satellite Network. In *proc. of International Conference on Communications* (June 2004), vol. 6, pp. 3321–3325.

[5] ETSI EN 301 790 v1.4.1. Digital Video Broadcasting (DVB): Interaction channel for satellite distribution system. September 2005.

[6] Henderson, T., and Katz, R. Network simulation for leo satellite networks, 2000.

[7] Institute, I. S. The network simulator - ns2. http://www.isi.edu/nsnam/ns/.

[8] K. Fall and K. Varadhan. Satellite Networking in ns, April 2006. the ns-manual, chapter 17.

[9] M. Karaliopulos, R. Tafazolli and B. Evans. An ns-derived GEO satellite network simulator: features, capabilities, results. In *in proceedings of Seminar Simulation Modeling Satellite System* (2002).

[10] N. Celandroni, E. Ferro, and Francesco Potortí. A simulation tool to validate and compare satellite TDMA access schemes. *Telecommunication Systems 12*, 1 (1999), 21–37.

[11] N.Bonelli, S.Giordano, G.Procissi and R.Secchi . BRUTE: A High Performance and Extensible Traffic Generator. In *proceedings of SPECTS'05* (July 2005).

[12] Potortì, F. Free software and research. In *proceedings of the International Conference on Open Source Systems (OSS)* (July 2005), pp. 270–271. Short paper.

[13] S. Pratt, R. Raines, C. F., and Temple, M. An operational and performance overview of the iridium low earth orbit satellite system. *IEEE Communication Surveys 1*, 3 (1999).